REMARKS

Claims 1, 2, 13, 25 and 26 have been amended. Claim 22 has been cancelled. Claim 37 has been added. Therefore claims 1-21 and 23-37 remain pending in the application. Reconsideration of the present case is earnestly requested in light of the following remarks.

Section 102(b) Rejection:

The Examiner rejected claims 1-6, 8, 9, 11, 13-18, 20, 21, 23-30, 32, 33 and 35 under 35 U.S.C. § 102(b) as being anticipated by Ramkumar et al. (U.S. Patent 6,154,877) (hereinafter "Ramkumar"). Applicants respectfully traverse this rejection for at least the following reasons.

Regarding claim 1, Ramkumar fails to disclose **deploying a plurality of application components on a server**. With regard to this feature of claim 1 (and all of the others), the Examiner cites column 11, lines 39-44 and asserts, "Ramkumar et al., teaches (col. 11, lines 39-42) a portable checkpointing method that provides a mechanism to mark potential checkpoint location, He thus teaches limitations pertain to deploying a plurality of application components on a server, determining checkpoint information for each application component" [*Sic*]. As the Examiner is certainly aware, anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). **The identical invention must be shown in as complete detail as is contained in the claims**. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Simply because Ramkumar "pertains to" the limitations of claim 1, does not mean he discloses the specific methods implemented by the invention as claimed.

With regard to the specific citation provided by the Examiner, Applicants note that column 11, lines 30-45 recites:

> In general, at compile time it is not possible to determine the ideal location for each checkpoint. However, a pre-compiler may be able to determine good potential checkpoint locations with a view to minimize the checkpoint size. At run time, an optimal checkpointing frequency can be computed and used to selectively exercise some of these checkpoints in the event that the potential checkpoints are encountered relatively frequently.
> In accordance with one embodiment of the invention, a portable checkpointing method provides (1) a mechanism to mark potential checkpoint locations, (2) timer-based control for selective activation of checkpoints. The first potential checkpoint marker encountered after the timer has expired is selectively activated. The timer is then reset following completion of the checkpoint.

Thus, in Ramkumar, a pre-compiler may determine locations for checkpoints in a program. **Applicants assert that one skilled in the art easily understands that using a pre-compiler for a single program does not teach deploying a plurality of application components on a server, as recited in claim 1.** Instead, Ramkumar only teaches that checkpoints may be inserted into a single program and that during run-time the checkpoints of that single program may be activated according to a timer. Ramkumar makes no mention or suggestion that indicates the plurality of application components recited in the claim. As stated above, the Examiner must show that Ramkumar teaches every limitation of claim 1; clearly, this portion (or any other of Ramkumar) fails to disclose deploying a plurality of application components on a server as recited in the claim. Thus, for at least the reasons above, Ramkumar fails to teach this feature of claim 1.

With further regard to claim 1, Ramkumar fails to disclose **determining checkpoint information for each application component, wherein said determining the checkpoint information comprises applying a checkpoint selection heuristic to an application component, wherein said applying the checkpoint selection heuristic comprises selecting checkpoints based on a description of one or more methods of the application component**. As argued above, Ramkumar fails to teach *deploying a plurality of application components on a server*; therefore, he cannot teach *determining checkpoint information for each application component* as recited in the claims. However, Applicants note that the Examiner cites the same portion as above as well as column 3, lines 1-5, with regard to this limitation. Applicants assert that **the cited portion of Ramkumar makes no mention of checkpoint information much less the specific determination recited in the claims.** In fact, Ramkumar provides very little details as to how the precise locations of the checkpoints are determined by the pre-compiler. For the most part, Ramkumar teaches that the user may insert calls that invoke checkpoints. For example, in column 12, lines 45-47, Ramkumar discloses "In the current embodiment, the user must specify potential checkpoint locations by inserting a call to the library function checkpoint". Applicants assert that such manual insertion of potential checkpoint locations is not the determining recited in the claims that comprises applying a checkpoint selection heuristic to an application component, wherein said applying the checkpoint selection heuristic comprises selecting checkpoints based on a description of one or more methods of the application component. Additionally, Applicants assert that Ramkumar nowhere discloses the specific methods described in this limitation. More specifically, Ramkumar makes nowhere teaches applying a checkpoint selection heuristic to an application component which comprises selecting checkpoints based on a description of one or more methods of the application component to determine checkpoint information for the application component.

With specific regard to the citation of column 11, lines 32-35, Ramkumar only discloses that a pre-compiler may be operable to determine checkpoint locations, not that it employs the specific methods recited in the claims. Simply because checkpoint locations are determined does not mean Ramkumar teaches or suggests *a checkpoint selection heuristic to an application component which comprises selecting checkpoints based on a description of one or more methods of the application component.* Applicants note that column 3, lines 1-5 discloses: "where the compiler identifies points in the program where checkpoints may be potentially taken and heuristics are used to determine which of these checkpoints will be activated". However, this method is prior art for Ramkumar; correspondingly, this heuristic does not apply to the system taught in Ramkumar and does not teach all of the limitations of claim 1. **Additionally, this citation fails to teach this particular feature of claim 1; more specifically, it fails to teach that the heuristic selects checkpoints based on a description of one or more**

methods of the application component. For at least the reasons above, Applicants assert that Ramkumar fails to teach this feature of claim 1.

Furthermore, Ramkumar fails to disclose **selectively checkpointing each component during execution of each component, according to the checkpoint information for each component**. As argued above, Ramkumar fails to teach the plurality of components and the determination of the checkpoint information recited in the claims and therefore cannot teach this feature of claim 1. However, Applicants notes that the Examiner cites column 11, lines 40-44 where Ramkumar teaches that checkpoints may be activated after each expiration of a time limit (recited above). Applicants assert that activating checkpoints after time intervals fails to teach, suggest, *or even hint at* selectively checkpointing each component during execution of each component, **according to the checkpoint information for each component**. As argued above, Ramkumar nowhere teaches the checkpoint information recited in the claims; additionally, Applicants note that the time interval is determined based on system properties and not the potential checkpoints. More specifically, in column 11, lines 25-29 Ramkumar discloses: "The optimal frequency of checkpointing depends on the underlying system's MTBF" (mean time between failures)... assuming that potential checkpoint locations are reached relatively often during MTBF, the minimum time between checkpoints (MinTBC) can be optimized based on the system's MTBF". Thus, the frequency of checkpointing (the time interval) depends on the underlying system's MTBF. Correspondingly, Applicants assert that Ramkumar fails to teach that the time interval depends on the checkpoints, much less the checkpoint information for each component recited in the claims. Thus, for at least the reasons above, Ramkumar fails to disclose *selectively checkpointing each component during execution of each component, according to the checkpoint information for each component* as recited in the claims.

Thus, for at least the reasons above, the rejection of claim 1 and those claims dependent therefrom, is not supported by the cited art and removal thereof is respectfully requested. Claim 25 includes similar limitations as claim 1 and so the above arguments apply with equal force to this claim. Thus, for at least the reasons provided above, Applicants submit that the rejection of claim 25, and those claims respectively dependent therefrom, is not supported by the cited art and removal thereof is respectfully requested.

Regarding claim 2, Ramkumar fails to disclose **wherein said applying the checkpoint selection heuristic to each application component comprises determining which of the one or more methods of the application component modifies the state of the component based on the description of the one or more methods**. Similar to arguments above, Ramkumar fails to teach that the determination comprises applying a checkpoint selection heuristic which comprises determining which of the one or more methods of the application component modifies the state of the component as recited in the claims. Ramkumar nowhere mentions or suggests *determining checkpoint information comprising determining which methods of the application component* *modifies the state of the component* as recited in the claims. Thus, for at least the reasons above, the rejection of claim 2 and those claims dependent therefrom, is not supported by the cited art and removal thereof is respectfully requested. Claim 26 includes similar limitations as claim 2 and so the above arguments apply with equal force to this claim. Thus, for at least the reasons provided above, Applicants submit that the rejection of claim 26, and those claims respectively dependent therefrom, is not supported by the cited art and removal thereof is respectfully requested.

Applicants also assert that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

Allowable Subject Matter:

Claims 7, 10, 12, 19, 22, 31, 34, and 36 were objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all limitations of the base claim and any intervening claims. Claim 13 has been amended to include the allowed subject matter of claim 22. Accordingly, claims 13-21, 23 and 24 are in condition for allowance. Also, added claim 37 is original claim 7 rewritten in independent form including all limitations of the base claim and intervening claims. Accordingly, claim 37 is in condition for allowance. All other pending claims should be in condition for allowance for at least the reasons discussed above.

CONCLUSION

Applicants submit the application is in condition for allowance, and prompt notice to that effect is respectfully requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above-referenced application from becoming abandoned, Applicants hereby petition for such an extension. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-66900/RCK.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard
☐ Petition for Extension of Time
☐ Notice of Change of Address
☐ Other:

Respectfully submitted,

Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: September 12, 2006